

314 Appendix

315 Part I

316 Table of Contents

318	A More Method Details.	10
319	A.1 Pseudocode of CARE.	10
320	A.2 CARE with OpenMask3D.	11
321	A.3 CARE with VLMaps.	11
322	B Additional Experimental Details.	11
323	C Limitations and Future Works.	12

327 A More Method Details.

328 A.1 Pseudocode of CARE.

329 To further enhance the reproducibility of our work, we present CARE’s pseudocode in Algorithm 1.
330 Additionally, we will open-source the code once our work is accepted.

Algorithm 1 Pipeline of CARE

```

1:  $S = \text{SelectionMethod} \in \{\text{TopConfidence}, \text{TopPrediction}\}$ 
2:  $k = \text{The boundary of ranking for the selection method}$ 
3:  $U = \text{UncertaintyMeasure} \in \{\text{Entropy}, \text{StandardError}, \text{KL}\}$ 
4:  $N = \text{Number of all the points}$ 
5:  $f = \text{Feature dimensions}$ 
6:  $m = \text{Number of all possible classes}$ 
7:  $O = \text{Set of every point, shape } (N, f)$ 
8: if original plan fails then
9:   if  $S == \text{TopConfidence}$  then ▷ Filter with selection method
10:     $C = \text{score of } O \text{ from high to low, shape } (N, 1)$ 
11:     $O' = \text{points with top } k \text{ highest scores, shape } (N', f)$ 
12:   else if  $S == \text{TopPrediction}$  then
13:     $P = \text{predicted class from high to low for each point in } O, \text{ shape } (N, m)$ 
14:     $O' = \text{points where their top } k \text{ predicted classes include the target class, shape } (N', f)$ 
15:   end if
16:   if  $U == \text{Entropy}$  then ▷ Choose a point with uncertainty measure
17:     $E = \text{entropy of } O', \text{ shape } (N', 1)$ 
18:     $O^* = \text{argmax}(E), \text{ shape } (1, f)$ 
19:   else if  $U == \text{StandardError}$  then
20:     $SE = \text{standard error of } O', \text{ shape } (N', 1)$ 
21:     $O^* = \text{argmin}(SE), \text{ shape } (1, f)$ 
22:   else if  $U == \text{KL}$  then
23:     $KL = \text{KL divergence of } O', \text{ shape } (N', 1)$ 
24:     $O^* = \text{argmin}(KL), \text{ shape } (1, f)$ 
25:   end if
26:   return  $O^*$ 
27: end if

```

331 A.2 CARe with OpenMask3D .

332 **Method Overview:** Following OpenMask3D [3], a transformer-based 3d instance segmentation
333 model is used to propose 3d masks. After the masks are proposed, up to 5 views where the object is
334 visible can be selected for calculating the mask feature.

335 **View Selection:** Given a 3d object mask, the 3d points in the mask are projected back to 2d for
336 all posed RGB images in the scene. We then validate whether the object is visible in a view by
337 checking if any of the points projected to 2d lies within the image. If there are more than 5 views
338 where the object is visible, we rank the images by the number of object pixels and choose the top
339 5 views. This not only helps us manage the computation cost but also encourages the selection of
340 views that are closer to the object, which might be helpful in filtering out far-away views that might
341 not capture the object clearly.

342 **Feature Extraction:** Following the original OpenMask3D implementation, we used CLIP-ViT-
343 L/14 for encoding images and texts. The 3d to 2d projection operation mentioned in the last para-
344 graph has also allowed us to calculate the bounding-box of the object which we refer to as "object
345 crops". For feature extraction, we encode the object crops with the CLIP visual encoder and save
346 them all instead of taking the average of them. In the original OpenMask3D implementation, they
347 also used multi-scale cropping for each object crop as a data augmentation. Since data augmentation
348 is orthogonal to the direction of this work, we omitted this part for simplicity.

349 A.3 CARe with VLMaps.

350 **Method Overview:** Following the original VLMaps [5], we first select 10 scenes and randomly
351 generate several poses, which includes position and rotation, with their corresponding RGBD obser-
352 vations. Then, the image features generated by LSeg[10] model are projected to the global frame.

353 **Map Generation:** As previously mentioned, we build the map by the method that is identical to
354 VLMaps. However, we further save some metrics for each grid which are utilized in our work, such
355 as entropy, standard error, and KL divergence. Additionally, to align with the method of feature
356 fusion [1] adopted in VLMaps, we calculate the above metrics in their weighted version.

357 **Navigation and Planning:** In the navigation stage, VLMaps first generate a mask indicating the
358 presence of a specific object class, and it then plans a path to the boundary of the nearest object.
359 With the provided path, it further calculate the angle and distance between two subsequent halfway
360 point, generating the low-level actions that is used in the HabitatSim [12, 13, 14].

361 **Evaluation and Re-proposing:** After all the actions are executed, we calculate the distance be-
362 tween the agent and the approximate boundary of the nearest object with the ground truth data
363 provided by the simulator. Following the settings in VLMaps, we count it success when the distance
364 is less than or equal to 1 meter. If it fails, we then generate a new proposal of where the object may
365 be by our method CARe. Similarly, we calculate the distance between the new point and its nearest
366 object, checking whether the distance is less than or equal to 1 meter.

367 B Additional Experimental Details.

368 **Quantitative Results:** In the setting of VLMaps, we also conduct the KL divergence method as
369 our uncertainty measure as shown in Table 3. However, some scenes with larger space or more data,
370 may cause the calculation of pairwise KL divergence become quite computational intensive. Thus,
371 we have to skip two larger scenes due to the limitation of memory size, which makes the result not
372 comparable to others.

Replan Strategy	k=4	k=8	k=16	k=40	k=100
No replan			54.4		
Max confidence (highest score)			67.0		
Random replan			56.2		
Min KL from topk confidence	82.7	83.1	85.2	84.5	82.7
Min KL from topk category	79.2	77.1	73.9	64.4	64.4

Table 3: VLMaps Replanning Subgoal Success Rates with KL divergence

Qualitative Results: We provide more qualitative results on the [anonymized project page¹](#), including the process of our proposed CARE solving an object navigation task. These qualitative results support our claims and make our work more convincing.

C Limitations and Future Works.

While our CARE effectively cooperates with existing pre-explored semantic maps and navigation models in a training-free manner to achieve better performance, it may be limited by one major assumption: CARE assumes that the navigation model has consistent decision biases. When the navigation model is updated, this decision bias may be eliminated or changed, resulting in less significant performance improvements that CARE can bring. In addition, because the structure of the semantic map may be different, CARE only uses the fixed semantic map for re-planning and does not further update the map with new information during the process. This research direction has the potential to continuously improve performance but is beyond the scope of this study. We will discuss and verify this direction in our future work.

¹<https://carmaps.github.io/supplements/>